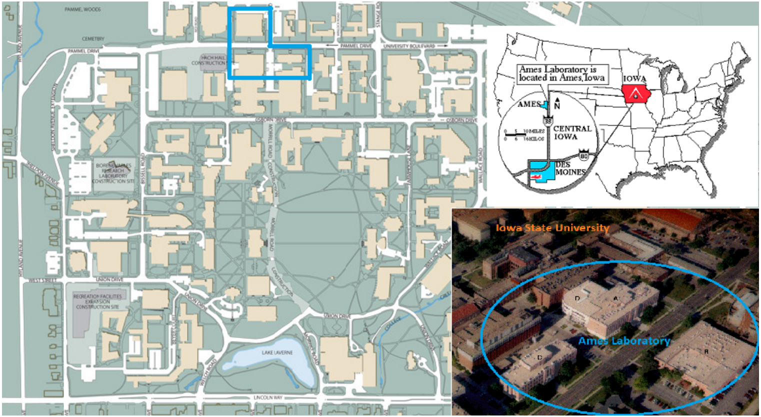
Dynamic Whitelist Generation for Automated Response

Chris Strasburg, Josh Adams

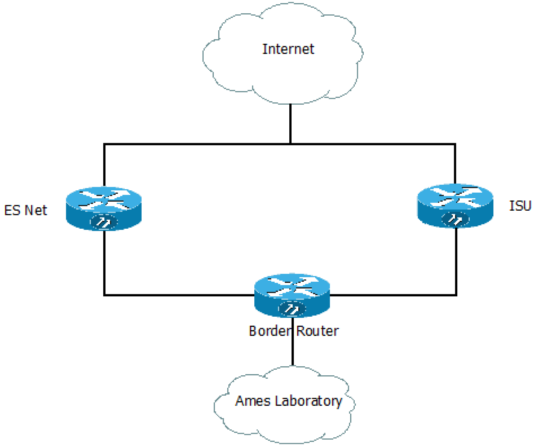
Ames Laboratory, US DOE
cstras@ameslab.gov, jadams@ameslab.gov

Outline

Ames Physical Environment



Ames Network Environment



Problem

Threats

- Effectiveness of attack tools
- Highly motivated attackers
- Deviation from classical attack sequence

Mitigation

- Near real-time information sharing
- Automated response included in tools

New Risks

- Unintentional Disruption
- Denial of Service

Whitelist Characteristics

Whitelists:

- Specify site-critical resources
- Prevent automated response block

Examples

- E-Mail Anti-spam Whitelists
- Web-filter / Proxy Whitelists

Challenges

- Environment specific
- Difficult to comprehensively construct
- Evolve over time

Other Options

- Post-response cost estimation

Goal: Utilize available data to automate whitelist generation.

Whitelist Characteristics

Whitelists:

- Specify site-critical resources
- Prevent automated response block

Examples

- E-Mail Anti-spam Whitelists
- Web-filter / Proxy Whitelists

Challenges

- Environment specific
- Difficult to comprehensively construct
- Evolve over time

Other Options

- Post-response cost estimation

Goal: Utilize available data to automate whitelist generation.

Whitelist Characteristics

Whitelists:

- Specify site-critical resources
- Prevent automated response block

Examples

- E-Mail Anti-spam Whitelists
- Web-filter / Proxy Whitelists

Challenges

- Environment specific
- Difficult to comprehensively construct
- Evolve over time

Other Options

- Post-response cost estimation

Goal: Utilize available data to automate whitelist generation.

Whitelist Characteristics

Whitelists:

- Specify site-critical resources
- Prevent automated response block

Examples

- E-Mail Anti-spam Whitelists
- Web-filter / Proxy Whitelists

Challenges

- Environment specific
- Difficult to comprehensively construct
- Evolve over time

Other Options

- Post-response cost estimation

Goal: Utilize available data to automate whitelist generation.

Whitelist Characteristics

Whitelists:

- Specify site-critical resources
- Prevent automated response block

Examples

- E-Mail Anti-spam Whitelists
- Web-filter / Proxy Whitelists

Challenges

- Environment specific
- Difficult to comprehensively construct
- Evolve over time

Other Options

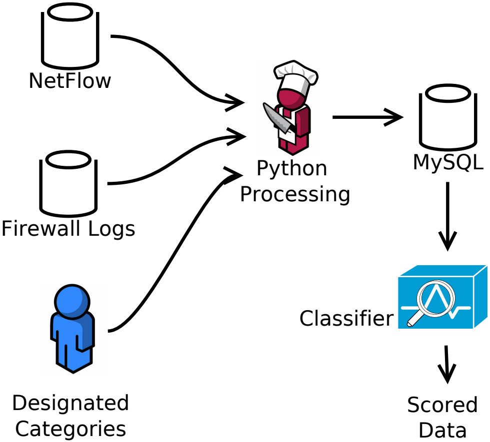
- Post-response cost estimation

Goal: Utilize available data to automate whitelist generation.

Desired Qualities (What makes a good whitelist?)

- Low false positive rate (whitelist an IP erroneously)
- Low false negative rate (fail to whitelist an important IP)
- Real-time classification
- Minimal maintenance
  - Updatable / adaptive
  - Self-generating
  - Intuitive scoring
- Easy to interpret

Dynamic Whitelist Architecture



Cooking Data

NetFlow

- Tuples of the form $(SIP, SP, DIP, DP, Bytes, Proto, Time)$
- Processed into aggregate statistics, and normalized
  - Number of flows
  - Number of bytes
  - Time since last visited
  - Peer count

Firewall

- Tuples of the form $(SIP, SP, DIP, DP, \{allow|block\}, Time)$
- Processed into aggregate statistics, and normalized
  - Number of blocks

(IP, flows, datavol, peercount, lastseen, blocks)

IP	flows	datavol	peercount	lastseen	blocks
aaa.bbb.ccc.ddd	1.85e-07	5.53e-09	0.00e+00	0.359	0
www.xxx.yyy.zzz	1.85e-07	3.71e-09	0.00e+00	0.711	12

Cooking Data

NetFlow

- Tuples of the form $(SIP, SP, DIP, DP, Bytes, Proto, Time)$
- Processed into aggregate statistics, and normalized
  - Number of flows
  - Number of bytes
  - Time since last visited
  - Peer count

Firewall

- Tuples of the form $(SIP, SP, DIP, DP, \{allow|block\}, Time)$
- Processed into aggregate statistics, and normalized
  - Number of blocks

(IP, flows, datavol, peercount, lastseen, blocks)

IP	flows	datavol	peercount	lastseen	blocks
aaa.bbb.ccc.ddd	1.85e-07	5.53e-09	0.00e+00	0.359	0
www.xxx.yyy.zzz	1.85e-07	3.71e-09	0.00e+00	0.711	12

Data Summary

- Collected one month of NetFlow data: $1,677,720$ IP addresses.
- Of those, $5,571$ were categorized according to our groups.

Category	Class	Count	Prior
ANL Whitelist Upstream Routers ISU DNS servers	w	36	1
ESNet Google Yahoo Search	l	754	0.75
.edu,.gov,.mil	d	$1,289$	0.5
Emerging Threats	b	$2,603$	0.25

Prior: $\approx$ P(Whitelist | Intrusion Response, NetFlow Traffic Seen)

Approaches

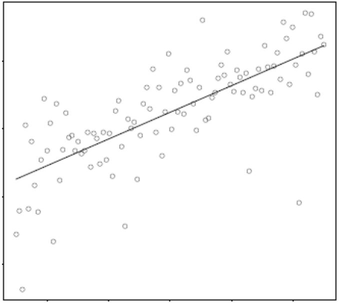
- Classifiers vs. Score Estimators
  - Classification = Score + Threshold
- Score-based dynamic whitelist approaches
  - Naive approach - any destination IP from an AL host
    - $28,908$ collected IP addresses were listed by dynamic blacklisting services.
  - Linear Regression - Linear function value estimate
  - Naive Bayes - Relative score
  - Bayesian networks - Probabilistic score

Approaches

- Classifiers vs. Score Estimators
  - Classification = Score + Threshold
- Score-based dynamic whitelist approaches
  - Naive approach - any destination IP from an AL host
    - $28,908$ collected IP addresses were listed by dynamic blacklisting services.
  - Linear Regression - Linear function value estimate
  - Naive Bayes - Relative score
  - Bayesian networks - Probabilistic score

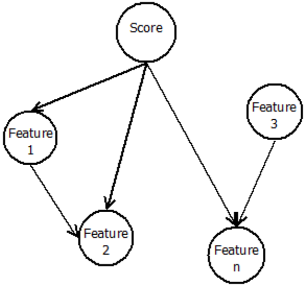
Linear Regression

- Fit a curve to data
- Model:
  $Y_i = \beta_0 + \beta \mathbf{x}_i + \varepsilon_i$
- Useful for:
  - Estimating fit of a model
  - Predicting values
- Pitfalls
  - Assumes linear model is appropriate
  - Assumes normal distribution
  - Assumes common variance between x_i

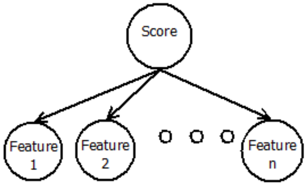


Bayesian Networks

- Based on Bayes' equation: $P(A|B) = \frac{P(B|A)*P(A)}{P(B)}$
- Bayesian Network: A graphical model of dependencies
- Useful for:
  - Modelling arbitrary distributions
  - Bayes Nets: Returning an actual probability
- Pitfalls:
  - Naive Bayes: Assumes all features are independent



Bayesian Network Model

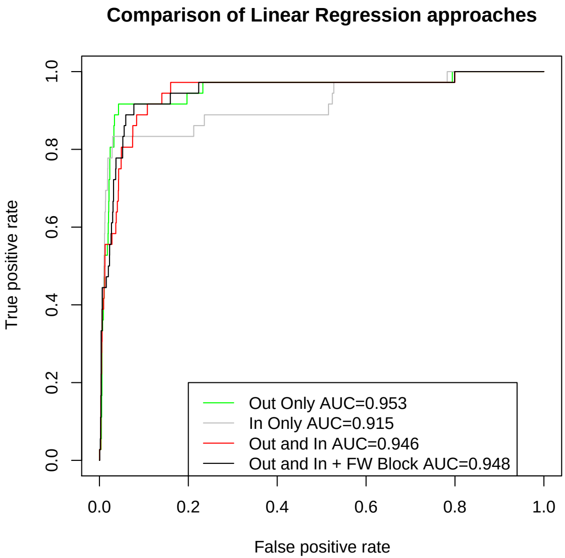


Naive Bayes Model

Tested

- Impact of Flow Directionality (In? Out? Both?)
- Feature Selection
- Score Type
- Bayesian Networks - Learned vs. Defined structure

Results: Regression



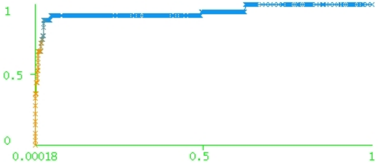
Results: Linear Regression Models

Out	$S =$	$F + L + D - F * L$
In	$S =$	$F + L - P - F * L$
In + Out	$S =$	$iL + oF + oD - iP + oL - oP - oF * iP - oF * oL + iL * oP - iP * oP - oL * oP - iL * oL - oD * oP$
In + Out + Block	$S =$	$iF + iL + oF - iP - oL + oD - iF * oF - oF * oL - iP * oD - iL * oL + iL * oF$

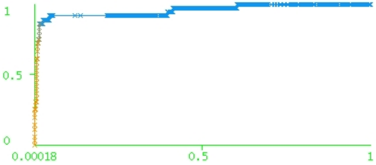
Legend:

S - Score	F - Flows
L - Lastseen	D - Data Volume
P - Peer Count	o - Outbound
i - Inbound	

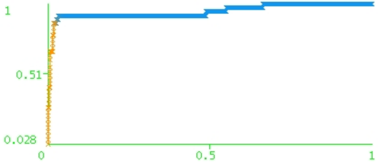
Results: Naive Bayes



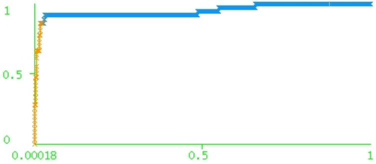
Inbound Flows Only (0.942)



Outbound Flows Only (0.953)

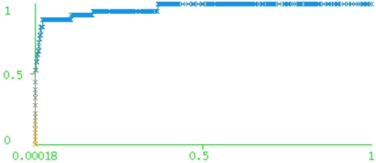


In and Outbound Flows (0.945)

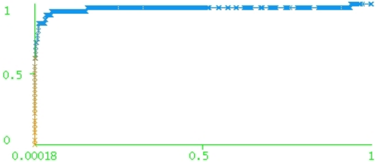


In and Out + Blocks (0.945)

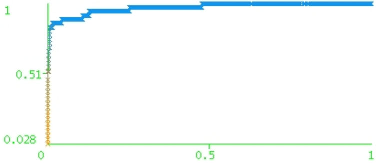
Results: Bayesian Networks



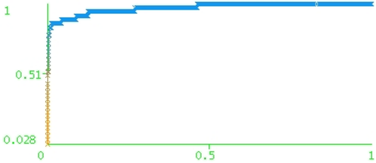
Inbound Flows Only (0.963)



Outbound Flows Only (0.967)

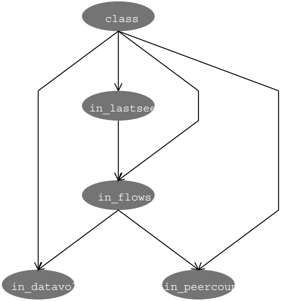


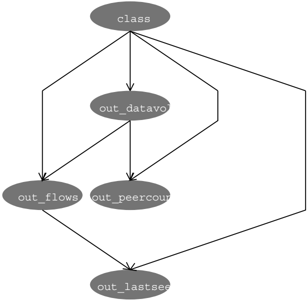
In and Outbound Flows (0.969)



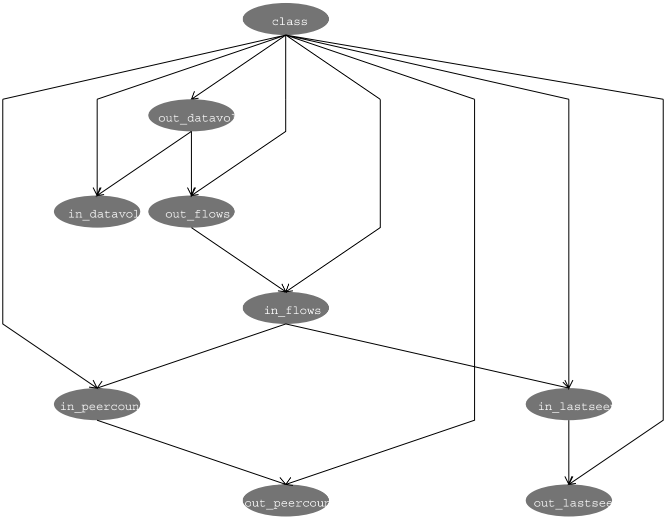
In and Out + Blocks (0.970)

Results: Bayesian Network Structures - In/Out Only

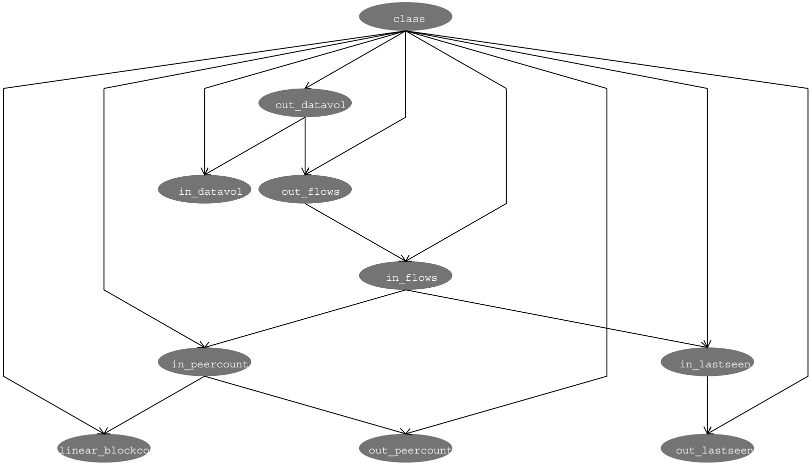




Results: Bayesian Network Structures - In + Out



Results: Bayesian Network Structures - In + Out + Block



Discussion / Speculation

- Linear Regression and Naive Bayes both have assumptions which do not hold.
- Coupling with post-response estimation may be beneficial.
- Sufficient information to model IPs seen frequently.

Method	Data	AUC
LR	Out Only	0.953
	In Only	0.915
	In + Out	0.946
	In + Out + FW	0.948
NB	Out Only	0.953
	In Only	0.942
	In + Out	0.945
	In + Out + FW	0.945
BN	Out Only	0.963
	In Only	0.967
	In + Out	0.969
	In + Out + FW	0.970

Take Aways

- Process to define a dynamic whitelist:
  1. Define broad categories of resources.
  2. Provide rough estimates of “priors”.
  3. Define features.
  4. Gather data.
  5. Apply classifier(s) using tool of choice. (Weka, R, Python, Matlab, etc. . .)
  6. Compare results with employed blacklists.
- Feasible to model important site resources with minimal effort / maintenance.

Continuing Efforts

- Live “burn-in” (AL Deployment)
- Address with changing behavior
  - Sliding window?
  - Time-based decay?
- Performance tweaking
  - Additional Features / Data Sources
  - Alternative scoring functions
- Generalize to other data types
- Explore other scoring approaches
- Portable tool

Acknowledgements

- Ames Laboratory
- NSM Group
- Argonne National Lab

References

- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.
- R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.
- Tobias Sing, Oliver Sander, Niko Beerenwinkel and Thomas Lengauer (2009). ROCR: Visualizing the performance of scoring classifiers.. R package version 1.0-4. http://rocr.bioinf.mpi-sb.mpg.de/
- David A. James and Saikat DebRoy (2008). RMySQL: R interface to the MySQL database. R package version 0.6-1. www.mysql.com www.omegahat.org bioconductor.org/packages/release/extra
- Stuart Russell, Peter Norvig (2003). Artificial Intelligence: A Modern Approach. ISBN 0-137903-95-2.